Parking Negotiation

1. Files in parking negotiation project
   A) config.py
      This file sets configuration parameters from cfg/sim_setting.ini file.
   B) constants.py
      This file sets constant values.
   C) datacenter.py
      This file defines Data Center class manages information of vehicles and parking lots (However in the current, data center don't manage information of parking lots).
   D) networkutility.py
      This file defines some methods to convert sumo network into network, which can be searched the shortest path with weights of a road, or search the alternative route.
   E) parkinglot.py
      This file defines Parking Lot class manages a parking lot, and get the state of parking lot.
   F) simulator.py
      This file defines Simulator class run server-side sumo's programs with some settings.
   G) testsimulation.py
      This file defines Test Simulation class overriding Simulation class implement park behavior of vehicles and whether vehicles should negotiate with each other at every simulation steps.
   H) anticipatory_stigmergy.py (not used)
      This file defines the strategy using anticipatory stigmergy, and is created by Mr. Jun.
   I) long_short_stigmergy.py (not used)
      This file defines the strategy using past stigmergy, and is created by Mr. Jun.
   J) negotiation.py (not used)
      This file defines the strategy using negotiation function with past

stigmergy and anticipatory stigmergy, and is created by Mr. Jun.

K) stigmergy.py (not used)
This file defines Stigmergy class manages information of vehicles, and is created by Mr. Jun.

L) store_json.py (not used)
This file define Store class manages and reserve stigmergies by json file, and is created by Mr. Jun.

M) store.py (not used)
This file define Store class manages and reserve stigmergies by redis, and is created by Mr. Jun.

N) util.py (not used)
This file defines some useful methods, and is created by Mr. Jun.

O) cfg directory
The files in this directory define the configuration parameters and SUMO-GUI configuration.

P) network directory
The files in this directory are used when SUMO creates network. These files can be created by netconvert from OSM network file.

Q) output directory
The files in the directory is the output of the simulation (xml file). We can create statistic graph (we can see how to create in http://sumo.dlr.de/wiki/Tools/Visualization)

R) parking directory
The files in this directory define the set of value needed for modeling parking lot and defining vehicle's objective parking lot and park time. [network name]_parking.csv is parking information, and [network name]_vehicle.csv is vehicle information.

S) tools directory
The files in this directory defines the useful method simulates by SUMO. tools/parking/osm2sumo.py defines the method that converts OSM file into SUMO file with parking lot information. The other tools are created by Mr. Jun.

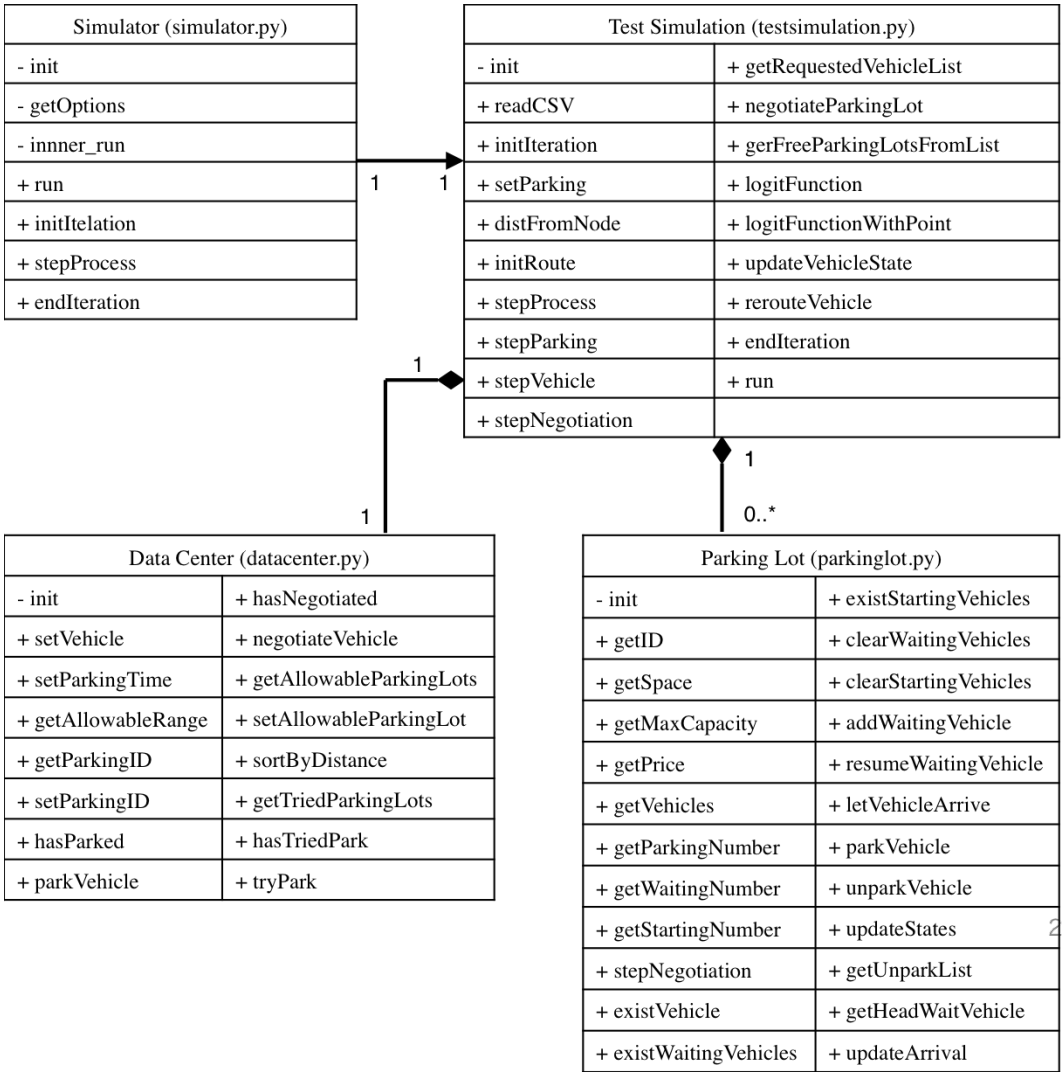
T) trip directory

The files in this directory are used when SUMO runs. These files defines each vehicle's route.

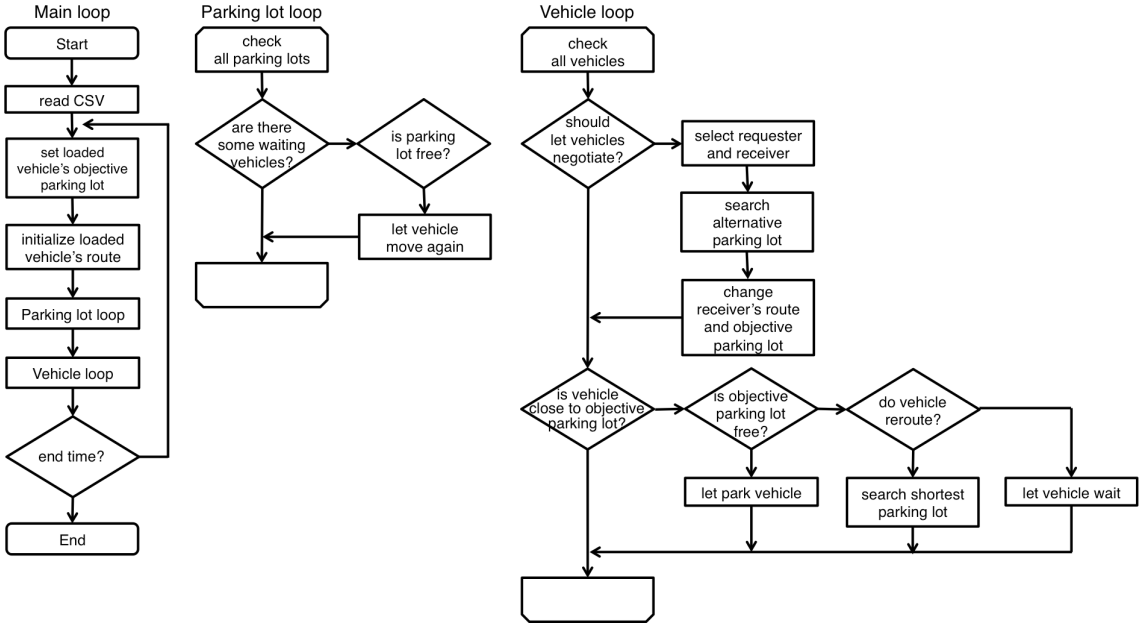
2. Configuration parameter

port：Port of traci
real_net：Flag of using osm network
parking：flag of using parking lot
reroute：flag of rerouting when objective parking lot is full
negotiation：flag of using negotiation function
iteration：iteration count
dce：flag of using distributed processing environment
dce_area：edge area of distributed processing environment (e.g. 2 means two edges)
redis_use：flag of using redis
redis_host：redis host. please set ip addresss or ' localhost'
long_term_sd：standard deviation of long-term stigmergy
short_term_sec：real time process interval (msec) h
weight_of_past_stigmergy：weight of long-term stigmergy
short_cut：in redis, which iteration is used ('-1' means not short_cut)
alpha：parameter in BPR funciton
beta： parameter in BPR funciton
congestion_bpr：parameter in BPR funciton
congestion_division：the ratio of vehicle assignment
congestion_judgement：the basis of conjestion in edge
requesterB：parameter in negotiaion
requestedB：parameter in negotiaion
point：parameter in negotiaion

3. Class diagram of parking negotiation system



4. Flowchart of parking negotiation



① read CSV

The test simulation reads CSV files defined in cfg/sim_setting.ini file. The parking lot object is created by parking lot information file (parking/[network name]/[network name]_parking.csv), and the data center store vehicle's acceptable range and parking time from vehicle's parking information file (parking/[network name]/[network name]_vehicle.csv).

② set loaded vehicle's objective parking lot

The loaded vehicle's objective parking lot is calculated from vehicle's acceptable range of parking lot. The data center collect information about loaded vehicle's objective parking lot.

③ initialize loaded vehicle's route

The test simulation initializes vehicle's route from vehicle's origin to objective parking lot, and from objective parking lot to vehicle's destination. The information of objective parking lot of vehicle is stored by data center.

④ let vehicle move again

If there are some waiting vehicles in the parking lot, and the parking lot is free, then the test simulation lets the waiting vehicles move again and aim at the parking lot.

⑤ select requester and receiver

If the data center that vehicle should negotiate another vehicles aiming at the same parking lot, the negotiation is conducted. In the negotiation, a requester, which requests another vehicle to select another parking lot, and a receiver, which receives the requester's request, are selected in the vehicles, which aims at same parking lot as the vehicle the data center judged it should negotiate. If there is no parking lot in vehicle's acceptable range, then the vehicle don't negotiate.

⑥ search alternative parking lot

The test simulation searches the alternative parking lot according to vehicle's acceptable range and distance between the parking lot and vehicle's destination. The alternative parking lot is selected the closest parking lot to vehicle's destination in vehicle's acceptable range.

⑦ change receiver's route and objective parking lot

The test simulation calculates the route from the current position to the alternative parking lot, and from the alternative parking lot to vehicle's destination. The data center updates vehicle's objective parking lot to the alternative parking lot.

⑧ let park vehicle

If there are some vehicles near the objective parking lot, and the objective parking lot is free, then the parking lot lets the vehicles park. Then the parking lot updates the information about parked vehicles and its capacity.

⑨ search shortest parking lot

If vehicle judged its objective parking lot is not free near its objective parking lot, then the vehicle reroutes. The test simulation searches the closest parking lot to vehicle's destination without vehicle's acceptable range and the parking lot that the vehicle has tried to park. The data center stores information about the previous objective parking lot as the parking lot that the vehicle has tried to park once.

⑩ let vehicle wait

If the vehicle don't reroute (cfg/sim_setting.ini file defines the flag of conducting reroute), the vehicle waits in front of the vehicle's objective parking lot. The waiting vehicles move again according to ④.

5. Goals

① Probability of negotiation

In this time, although probability of negotiation is 100%, we should calculate accurately by logit model. We may have to adjust parameter of logit model.

② Order of negotiation

In this time, order of negotiation is based on vehicle's ID. This causes that the priority of the vehicle which has lower ID becomes higher. We should change order of negotiation to random or another order.

③ Simulation in real situation

Although we use the simple scenario, when we simulate more complex scenario, there may be some errors and the result may become bad. Vehicle's demands should be changed to more real demands. We must construct data for parking negotiation simulation

④ Reroute behavior of vehicles

If vehicles find the objective parking lot is full in every time, the vehicles may have to park the parking lot, which is very far to vehicle's destination. That's why we should introduce the probability of selection whether the vehicles select to reroute or wait into our system. And to

consider on-street parking lot, we should consider vehicle's behavior to go back the parking lot and run around the area of the parking lot, even if the vehicle has arrived the parking lot. First, in order to represent this behavior, we should define on-street parking lot by the area and the route search also should be conducted according to the area. So, we need to define the area as a node. However, we must not create the node in SUMO network. Only when we conduct the route search, the node is used. Second, we should conduct route search in the area. As Prof. Miguel said before, I think we should search the route at random in the area. We need to define the acceptable range of reroute, and the vehicle should reroute in the acceptable range.

⑤ Information of parking lot

In this time, we consider only the number of vehicles when the data center judged the vehicle should negotiate. We should also consider the parking time of the parking vehicle in the parking lot. However, this is still no idea.